Stochastic shale permeability matching: Three-dimensional characterization and modeling

Pejman Tahmasebi a,b,⁎, Farzam Javadpour a, Muhammad Sahimi b

a Bureau of Economic Geology, Jackson School of Geosciences, The University of Texas at Austin, 78713 Austin, TX, USA
b Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, CA 90089-1211, USA

A R T I C L E   I N F O

Article history:
Received 20 June 2016
Received in revised form 22 August 2016
Accepted 26 August 2016
Available online 29 August 2016

Keywords:
Shale-gas
Characterization
Stochastic modeling
SEM image
Shale permeability

A B S T R A C T

The recently-discovered promise of shale-gas reservoirs necessitates their characterization. As a first step, two-dimensional (2D) scanning electron microscope (SEM) imaging can provide excellent view of the complexity associated with such reservoirs, and has been used in recent years to study organic materials, clays, minerals, and the pores in shales. Other important information, such as the connectivity of the pores in 3D and such macroscopic properties as the permeability are difficult to infer from 2D SEM images. Newer techniques, such as focused ion-beam SEM (FIB-SEM) have been utilized to overcome the shortcomings. The extremely small sample size and the costs associated with the FIB-SEM technique limit, however, wide use of the FIB-SEM in characterization of shale samples. An alternative approach is to use the recently developed advanced algorithms for 3D reconstruction that use one or a few 2D samples. Such methods may not, however, be able to fully reproduce the shale permeability. To accurately reproduce the permeability, we propose a new method based on a combination of the Metropolis-Hastings and the genetic algorithms. The new method learns from its own previously generated realizations of the shale and produces models that match the existing permeability data. The method is validated with the measured permeability for an actual 3D shale sample. It generates an ensemble of stochastic realizations that honor the permeability data, which may then be used for more accurate characterization of shale gas reservoir and analysis of their pore network.

Published by Elsevier B.V.

1. Introduction

Due to their special intrinsic physical characteristics, such as multimodal nano-scale structures and complex features, shale formations are of much interest in various studies in earth science, including geology, water resources, oil/gas reservoirs, CO2 sequestration, and many other related subsurface systems and phenomena. Owing to their immense hydrocarbon content as unconventional reservoirs, shales have recently become even more important. The size of the pores in shales is a distinct aspect of such formations that gives rise to complex morphology and connected flow paths. Such multiscale pore structures have considerable influence on the methodology and tools for studying shale reservoirs in terms of fluid flow, imaging, pore network modeling, and interpretation of the data (Javadpour, 2009; Sahimi, 2011).

Permeability is perhaps the most critical property for evaluating shale gas reservoirs. A precise evaluation of the permeability helps one to understand how the pore networks and important petrophysical parameters in unconventional reservoirs differ from those in the conventional ones. Unlike conventional reservoirs, gas flow in shale gas reservoirs is determined by various properties, including the multiscale structure of the reservoirs, their nano-porosity, and special flow mechanisms. Therefore, a considerable number of samples are needed to understand the variability and complexity of the features that determine the permeability.

Accurate 2D and 3D imaging is a prominent tool for better understanding of the complexities of shale gas reservoirs (Javadpour et al., 2012; Loucks et al., 2012). Such images provide excellent inside look into the depositional environment, mineralogy, maturation, thermal condition, total organic carbon (TOC), strain/stress properties, porosity, permeability, and the pore network of the formation. Thus, deeper studies of such reservoirs inevitably require a large number of samples using two- or three-dimensional (3D) images. Due to its use of various signals, high-resolution focused ion-beam scanning electron microscope (FIB-SEM) method (Lemmens et al., 2011) offers images that contain very useful information about the morphology and composition of shale samples (Loucks et al., 2012). Such images also provide a means for accurate evaluation of the complex connectivity in the pore network of shale reservoirs. Such key parameters as the TOC, mineralogy, porosity, permeability, and other petrophysical properties are best evaluated through the use of such images. The necessity of using several 3D samples to represent the heterogeneity and complexities in shale gas reservoirs was investigated by Kelly et al. (2015), who concluded that a vast number of samples are needed to improve the likelihood of a reliable assessment of the various petrophysical parameters. Obtaining such a large number of 3D images is, however, neither feasible nor practical.
On the other hand, 2D images can be provided with ease and at a low cost. Because of the obstacles involved with obtaining 3D images, Tahmasebi et al. (2012, 2015a) proposed a new approach based on taking one or more 2D images and building a 3D model stochastically. The models generated by their approach exhibit petrophysical and flow properties similar to those directly observed and measured in actual samples. Their method produces an ensemble of 3D realizations that offer acceptable approximation of the same properties in the 2D image(s). We should point out that, in addition to our method, there are several other important techniques that can reproduce the connectivities at small scales (see, for example, Adler et al., 1990; Roberts and Teubner, 1995; Yeong and Torquato, 1998a, 1998b; Jiao et al., 2009; Mehmani et al., 2013; Gerke and Karsanina, 2015; Gao et al., 2015). But, they have never been tested for reconstructing models of shale reservoirs.

Aside from recent progress in producing various equi-probable 3D realizations of shales (Tahmasebi et al., 2015a), reproduction of accurate permeability for complex structures remains an essentially open problem (Veselý et al., 2015; Gao et al., 2015). Indeed, it is perhaps excessive expectation of reproducing the existing experimental data using the current 3D stochastic models, when they are generated from one or very few 2D images. Intuitively, we know that 2D images can hardly convey, for example, a flow property that depends on the spatial pore connectivity in 3D. Furthermore, since one cannot study all the core data and samples due to the limitations of cost and time-consuming nature of the task, stochastic modeling can generate many samples, quantify the uncertainties, and put forward a range of variability. Clearly, the initial data—here the input 2D images—should represent the variability in the structures, in which case the stochastic methods produce various possible realizations in a matter of a few CPU seconds.

Building a 3D model of porous media based on limited data has been investigated extensively. Such models may be divided into three groups. In one group, called object-based techniques, the pore and grain structures are treated as a set of objects that are defined based on the prior knowledge of the pore space (Pyrcz and Deutsch, 2014). Clearly, such techniques are not applicable to random structures. Instead, the second group, called pixel-based methodologies, can be used effectively to produce the shapes that are hard to fit to a specific regular object (Stre´belle, 2002; Okabe and Blunt, 2004). Unlike the object-based methods, however, these techniques are unable to produce very realistic pore structures. Thus, one must resort to the third class of such techniques, called process-based models (Bryant and Blunt, 1992; Coelho et al., 1997; Biswal et al., 1999, 2007; Øren and Bakke, 2002) that produce very realistic models by mimicking the real processes that form the porous media. They are not, however, general enough as they are designed for specific sedimentary conditions appropriate for a given reservoir and, thus, need to be defined separately for every new type of reservoir. Furthermore, they also require highly intensive computations.

As a very effective alternative, a combination of the object- and pixel-based methods can produce more realistic morphology for porous media (Tahmasebi et al., 2012). In other words, the strength of pixel-based method for producing complex pore space and object-based methods for realistic models are integrated. For this aim, one of the recent techniques in this class of methods (Tahmasebi et al., 2015a) is implemented in the present study.

None of the previously described methodologies is capable of reproducing accurately the experimental data for the permeability and electrical conductivity. They are able to produce the spatial structure of the pore space of porous media, but reproducing the data for the flow and transport properties is not guaranteed. This is the reason why the stochastic methods produce an ensemble of realizations and, consequently, the petrophysical properties that are used for uncertainty quantification. They produce a range of, for example, permeability, instead of one single value. In this paper, a new iterative stochastic method for reproducing the permeability from 2D SEM images is proposed. Using concepts from the genetic algorithm (GA) and data mining, the new method uses various 2D images from different parts of a shale gas reservoir in order to iteratively improve the estimation of the permeability. To this end, the accepted realizations are pooled together to make a set of soft data (SD) on which the subsequent models are conditioned. The pool of the realizations and the SD are updated progressively, building upon the previously-accepted realizations. As with the optimization problems, the initial guesses (i.e. models) may produce results very different from the existing experimental data. At the later steps, however, as the pool of the realizations and the SD are updated, the calculated attributes draw closer to the actual measured experimental data. Thus, this paper presents a framework by which the experimental properties can be reproduced more or less exactly.

The rest of this paper is organized as follows. In the next section we discuss the methodology that we propose in this paper, including an algorithm that was recently suggested and utilized in the present paper, and how it is integrated with the GA. The method is then validated by two examples, a 2D synthetic model of shales and a 3D example based on an actual shale sample.

2. Methodology

Major obstacles to the development of shale reservoirs are their costly analysis and studies that point to the need for the development of more efficient technologies. As mentioned earlier, a basic and common type of study is conducting extensive analysis of various aspects of a large number of samples, including their mineralogy and some essential petrophysical and laboratory properties. Permeability is a key property that is evaluated for
providing a realistic assessment of fluid flow in shale formations. Therefore, the permeability data are usually available for many samples of shale gas reservoirs.

At the same time, detailed imaging provides a direct view of a shale reservoir. High-quality 3D images are usually not affordable, however, and thus there is always a trade-off between the cost and amount of information that can be acquired. Generally speaking, most methods of predicting the permeability are based on empirical equations that try to bridge the experimental data and the petrophysical parameters extracted from the SEM images. Such techniques usually lead to a single equation that tries to model the complexity and heterogeneity of shale reservoirs, which is, however, not realistic. In addition, such equations usually differ for different reservoirs, and cannot utilize all the information gathered, including the qualitative (cross-section maps, SEM images, etc.) and quantitative (core data, experimental results, etc.) data. Thus, important questions arise: (i) how can such data be integrated, as the quantitative experimental data for, for example, the permeability and the SEM images are entirely different, and (ii) how can the maximum amount of information be extracted from the SEM images and be anchored to the quantitative data, or vice versa? We will address the two questions and present a method that reduces the computation cost, using an efficient approach.

The methodology proposed in this paper is cast on a recently-developed method called the cross correlation-based simulation (CCSIM). The CCSIM algorithm (Tahmasebi et al., 2012) is an efficient stochastic approach that utilizes 2D/3D images and generates 3D models or realizations of a given porous formation. Then, the algorithm is integrated with an optimization and training framework to generate an ensemble of 3D realizations of shale that reproduce the experimental results. The realizations are used to investigate the flow properties of complex pore networks in shale-gas samples.

2.1. The CCSIM algorithm

The CCSIM algorithm is based on high-order statistics and, unlike the existing methods of shale characterization, extracts the maximum amount of information from an image or data set in order to reconstruct the shale samples. Indeed, lower-order statistical methods, such as covariance-based approaches, are unable to characterize the complex patterns and, consequently, cannot generate accurate models. Complex and multiscale patterns in unconventional reservoirs require methods that can reproduce such features. Aside from obtaining more information, however, it is also important to uncover the physical processes within the reservoirs that have contributed to the formation of their morphology and, thus, are also of great importance for accurate characterization of shales. The understanding that such physical processes and complexities produce are then translated into a digital image (DI) that is widely available from many shale samples.

Briefly, the CCSIM algorithm is similar to a concept in problem-solving in which a model is generated using the similarity between the pieces (i.e., the patterns) in the DI. In other words, since we are more interested in the uncertainty that exist in any such data due to lack of complete information, an exact match between the patterns in the DI and the model is not the goal of the CCSIM, rather generating multiple stochastic realizations of the same DI is pursued.

Mathematically speaking, most of the high-order statistical methods are based on the Euclidean distance, i.e. a function that characterizes the differences between the DI and a sample of the data taken from the DI. Computations with such a distance function is, however, is very costly. This motivated Tahmasebi et al. (2012) to propose a new methodology based on a cross-correlation function (CCF) for which the computation takes very little CPU time. For a complex DI represented by an image, the CCF yields much more accurate realizations of the image, because it takes into account its higher-order statistics. Physically, the CCF provides a measure of similarity to a dataset in a realization of the DI.

We describe the CCF and the CCSIM algorithm for 2D systems. Its extension to 3D media is straightforward. Let G represent the computational grid used in the modeling. The templates — the blocks of G — by which the DI is reconstructed are denoted by T, while D(T) is the data event at position u in T. Every two neighboring blocks share an overlap (OL) region. Use of “data event” instead of “data” is preferred because during the modeling, D(T) changes as the algorithm progresses. Finally, the OL regions represent the overlap segment between neighboring templates. Suppose that DI(x,y) represents the datum at point (x,y) of a DI of size Tx×Ty with x ∈ {0,...,Tx−1} and y ∈ {0,...,Ty−1}. Examining the DI, we focus on a portion D(T) of size OL×OL and regenerate it based on the data such that it matches the corresponding portion in the DI. The matching is based on the CCF, defined by

\[
C(i,j) = \sum_{x=0}^{OL-1} \sum_{y=0}^{OL-1} DI(x+i,y+j)DI(x,y) \tag{1}
\]

Where DI(x,y) is the data event at point (x,y). Eq. (1) indicates that the desired position of (i,j)—the best match with the DI—is one that maximizes C(i,j). If the DI is not too large - typically images smaller than 700 × 700 pixels - all the computations are carried out in the spatial domain (Tahmasebi et al., 2012; Tahmasebi and Sahimi 2016a,b). When the DI is very large, the calculations of the CCF, which is between the DI and D(T) in the Fourier space reduce the CPU time very significantly (Tahmasebi et al., 2012). The CCSIM algorithm is illustrated in Fig. 1, which shows how a new pattern is inserted in G.

2.2. Three-dimensional reconstruction

The CCSIM, in its original form, can be used for direct 2D-to-2D and 3D-to-3D modeling. For example, it can generate a 3D model using a 3D DI. In what follows, we describe an extension of the CCSIM algorithm that uses a few 2D DIs to reconstruct a 3D model. Intuitively, 3D (p × q × r) porous media may be thought of as a series of r 2D planes of size (p × q) that are stacked together in a given direction. In most stationary porous media (Tahmasebi and Sahimi, 2015a), one can identify a “vertical” connectivity perpendicular to the 2D images that are used. Thus, one can develop models of 3D porous media by reconstructing many interdependent 2D layers (planes) using the CCSIM algorithm and the given DI, and then stack them together to form the final 3D model. To preserve the external continuity, we first reconstruct the orthogonal external frames using the CCSIM and the DI, and then fill in the internal part of the system layer by layer. In this method, the first layer is reconstructed using the DI (in fact, we can use the 2D DI itself as the first layer), and reconstruct the rest layer by layer using the conditional CCSIM, one in which some of the data must be honored exactly. This means, in the present context, that each time we reconstruct a new layer, some of the data in the previously-reconstructed layer must be honored exactly in order to preserve the connectivity between the two layers. Thus, to preserve the continuity of the system, the first layer must be conditioned to its four external vertical edges. Then, to reconstruct the second layer, it is conditioned to both the external...
frames and the previous layer (in this instance, the first layer). Conditioning in this case is performed through point data that are extracted from the previous layer. Such point data are identified using the Shannon entropy based on the entropy of the previous layer. These steps are iterated until the uppermost predefined layer is reached and the 3D model is complete (Tahmasebi et al., 2015a). It should be noted that using just one single 2D DI implies isotropy assumption in the porous medium. Anisotropy can still be considered if one uses more images in different directions. In such modeling, each 2D image represents a specific directional anisotropy. For the sake of simplicity, isotropy assumption is made in this study.
Our previous work (Tahmasebi et al., 2012) indicated that, though highly efficient and in many cases very accurate, the method might generate some artifacts if the DIs are very complex and host multiscale features. To alleviate this problem and increase the quality of the realizations, we integrate an iterative algorithm with the CCSIM in such a way that it removes the unwanted patchiness and discontinuities (Tahmasebi et al., 2015; Tahmasebi and Sahimi, 2015b). First, a 3D realization is generated using the CCSIM algorithm that may contain some artifacts and patchiness. The patchiness is identified using the mismatches between the overlaps when two of them are interlocked in the 3D model. Then, based on the initially generated realization, the voxels of the 3D simulation grid are visited one-by-one. Three perpendicular 2D planes of size \((T \times T)\) are extracted at each voxel. Then, the similarity between each extracted plane and the DI is evaluated using the CCF. A set of candidate patterns are then identified for each plane. Next, the final value of the visiting voxel is defined based on the Jensen-Shannon equation (see below). To this end, the histograms of the 3D model for all of the candidate patterns are constructed and their distances to the DI are calculated. Finally, the value of the pattern that minimizes the aforementioned distance is selected and inserted in the 3D model. This algorithm iteratively removes the patchiness and artefact in the original 3D model (Tahmasebi et al., 2015b).

2.3. Workflow

Several factors affect performance of shales and its forecasting. Contributing to the uncertainty is lack of extensive data and accurate analysis of the existing samples. Improving our understanding of such factors reduces investment risks, leading to more accurate forecasts. Thus, having a set of acceptable realizations of the shale samples that have the same permeability as the actual ones provides a reliable view of such complex reservoirs. Another advantage of having satisfactory realizations is their usefulness in assessing the uncertainty and the connectivity of the macroscopic pore network that strongly affects gas production from shales. Analyzing a large number of realizations is not, however, feasible, and indeed, is very costly.

The various data sets available for shale reservoirs make it difficult to establish a generic workflow that can efficiently integrate them. For example, core and seismic data in such reservoirs provide valuable information about the porosity, mineralogy, petrophysical properties, and large-scale structures. The multivariate CCSIM can be used to integrate such data with the shale-gas models. Integration of the experimental data for such properties as the permeability and electrical conductivity requires, however, an iterative process that includes generating various samples and evaluating the important parameters using forward modeling. The iterative process is very demanding computationally, however, and is not similar to and as systematic as an optimization algorithm. In addition, producing a large number of realizations and forward modeling is tedious, even computationally prohibitive, taking days or weeks. Therefore, keeping low the number of required realizations and, consequently, the number of forward simulations is critical. Moreover, due to the nonlinear relationship between the shale samples and the responses of their forward modeling, different models can produce very similar responses, creating uncertainty.

The objective here — to produce many realizations and then iteratively alter them to better match the experimental data for the directly-measured permeability and electrical conductivity — can be viewed as an inverse problem in which several realizations are first generated and then a few forward simulations are performed (Tarantola, 2005). One useful approach for this objective is through Bayes’ rule by which the uncertainty is quantified using a posterior distribution of acceptable realizations. The posterior distribution of the shale realizations may be thought of as the product of a likelihood function that measures the similarity between the responses of forward modeling and the experimental data. The existing information on the shale samples’ parameters is presented by the DIs. Note that Bayes’ formulation allows simultaneous integration of different types of data. Mathematically speaking, a distribution can be rebuilt when the entire space is explored, a process that can be thought of as generating numerous realizations that fall within the posterior distribution. Clearly, sampling complex, multiscale, and non-Gaussian properties of shales is not straightforward. Thus, another objective of this study is developing an efficient and “intelligent” sampling method for such posterior distribution.

To achieve our objective, we split the problem into a series of iterative processes that minimize the differences between the experimental and calculated data, while the generated realizations preserve the defined parameters of the existing information contained in the DIs. A combination of Markov Chain Monte Carlo (MCMC) (Robert and Casella, 2013), the genetic algorithm (GA) as a stochastic optimization tool, and a training algorithm converge to the global minimum of an energy function that indicates convergence to the true model. The combined techniques reduce the CPU time of the MCMC by reducing its high rejection rate. The final algorithm is efficient and the acceptance rate of the generated realizations increases gradually, while the CPU time decreases. Let us describe our method.

As stated, we use Bayes’ rule to update data on the shale properties based on the experimental data \(d_{\text{exp}}\) such as the effective permeability. It is also assumed that the existing information is integrated into the shale realizations that are generated using the CCSIM. Thus, in a Bayesian context, the posterior probability is expressed by:

\[
 f(m | d_{\text{exp}}) = \frac{f(d_{\text{exp}} | m) f(m)}{f(d_{\text{exp}})}
\]

(1)

or, more simply,

\[
 f(m | d_{\text{exp}}) \propto f(d_{\text{exp}} | m) f(m)
\]

(2)

where \(f(m | d_{\text{exp}})\) is the posterior probability distribution, \(f(d_{\text{exp}} | m)\) is the likelihood function, and \(f(m)\) indicates the prior probability distribution for the porous media. Solving Eqs. (1) and (2) entails drawing a considerable number of samples from the posterior distribution, which can be done when the other two components, the prior \(f(m)\) and the likelihood \(f(d_{\text{exp}} | m)\), are known. Sampling is not, however, straightforward, and for this reason the MCMC as an efficient sampling technique is employed to draw samples from \(f(d_{\text{exp}} | m)\).

In some cases, however, one may also take experimental errors into account. Assuming a Gaussian distribution for the prior model and the error, Eq. (2) is rewritten as:

\[
 f(m | d_{\text{exp}}) \propto \exp[-S(m)]
\]

(3)
where $k$ is a normalization constant, and $S(m)$ is the least-squares misfit function, defined by:

$$S(m) = \frac{1}{2} \left[ (f(m) - d_{exp})^T C^{-1}_{D} (f(m) - d_{exp}) + (m - m_{prior})^T C^{-1}_{m} (m - m_{prior}) \right]$$

(4)

where $C_D$ is the covariance of the experimental data, $f(m)$ is the response of the generated shale realization $m$ that must be obtained using forward modeling, $m_{prior}$ is the prior model or original DI, and $C_m$ is the covariance of shale samples' parameters. Note that the 3D model should be first generated since the original DI is in 2D. This part is further discussed in the next section.

The MCMC is a technique for sequentially drawing realizations $m$ from a probability distribution. The result after a number of iterations is used as a sample of the target distribution $f(m | d_{exp})$. A new realization is drawn based on the previously drawn medium in such a way that its equilibrium probability is the target distribution. The most important advantage of this method is its improvement of the equilibrium distribution at each step as it converges to the target distribution. This method is used in this study because the target distribution, $f(m | d_{exp})$, is not available, and the MCMC can be used to build or approximate it. Here, however, due to its efficiency, the Metropolis-Hastings (MH) algorithm, a member of the MCMC family, is used for sampling and approximating the target distribution (Metropolis et al., 1953; Hastings, 1970). With the MH method, we begin with generating one realization using the prior distribution data. Then, a candidate realization from the proposed distribution is generated as the next sample, and its acceptance probability is calculated using the proposed distribution and the full joint probability density.

Clearly, the MH algorithm is computationally expensive, requiring thousands of realizations to build the target distribution. The forward modeling is also excessively time consuming. Thus, the candidate realizations must be produced in a way that accelerates the convergence and increases the acceptance rate. For this objective, we use the CSSIM algorithm with its advantage in terms of using auxiliary data, i.e. soft data. Similar to the learning rules of neural networks, all the accepted realizations are placed in a pool from which the algorithm learns about the previous realizations and their responses, which makes it more productive. In addition, other concepts, such as mutation used in the GA, are also integrated. Mutation in the GA is a rule that if it doesn’t lead to a completely plausible model, one can use the 2D DIs to build the 3D realizations, but this increases the CPU time. Thus, as mentioned earlier, we use the multivariate CSSIM algorithm, which requires the DI, soft data (SD), and an auxiliary variable (AUX). Briefly, the CSSIM uses the SD, which in essence constitutes a probability map, as the conditioning data. The SD creates a continuous and smooth map that is different from the existing SEM images. Therefore, the SD and the DI must be linked in order to condition the output model. To construct such a link, we need more data in the form of an auxiliary map, which is generated using an inverse CSSIM, i.e. one in which we exchange the DI with the SD, and use the latter as the DI. In particular, the AUX data are the output of the CSSIM when the DI and the SD are switched. The algorithm commences by generating 3D realizations using the given 2D DIs. Next, forward modeling is performed, e.g. simulation of flow and electrical conduction. For each new realization, the computed responses in the presence of an error are examined using the following equation, by which the probability of accepting any new realization compared to the previously-accepted one is tested:

$$\rho(m_{n-1}, m_{n}) = \min \left\{ 1, \exp \left( -\frac{1}{2} \left[ (f(m_{n-1}) - d_{exp})^T C^{-1}_{D} (f(m_{n-1}) - d_{exp}) + (m_{n-1} - m_{prior})^T C^{-1}_{m} (m_{n-1} - m_{prior}) \right] \right), \exp \left( -\frac{1}{2} \left[ (f(m_{n}) - d_{exp})^T C^{-1}_{D} (f(m_{n}) - d_{exp}) + (m_{n} - m_{prior})^T C^{-1}_{m} (m_{n} - m_{prior}) \right] \right) \right\}$$

(5)

The candidate realization is accepted with probability $\rho$ or rejected with probability $1 - \rho$. The new model, $n$, is accepted if it provides a more accurate response than the previously accepted realization, $n - 1$. If it is accepted, the new 3D model is added to the pool of the realizations. Each accepted realization is given a weight using the following equation,

$$\omega = \exp \left[ -\left( (f(m_{i}) - d_{exp})^T C^{-1}_{D} (f(m_{i}) - d_{exp}) + (m_{i} - m_{prior})^T C^{-1}_{m} (m_{i} - m_{prior}) \right) \right]$$

(6)

The weight defines the proximity of the permeability and conductivity of a stochastic realization to the corresponding measured data. For example, the first few accepted realizations are given a lower weight, as the subsequently accepted ones provide a better match. As mentioned, some concepts of the GA, such as mutation, are also used to avoid trapping in a local minimum. Thus, the given weights for the SD are changed using a Gaussian distribution after some random steps. This provides a chance for exploring more scenarios, which is similar to the mutation operation in the GA. Assigning a weight can, however, be thought of as the crossover step in the GA in the sense that it combines the available realizations - the so-called “parent” in the GA - to make a new realization - the “children” in the GA. At the next step, the SD is updated using the new pool and weights. Clearly, the realizations with higher weights constitute a greater part of the SD. Accordingly, the AUX data are also updated using the new SD and the selected realizations. The DI is updated using the newly accepted pool of the realizations to ensure that the most informative models are used as the DI (see below). The steps are repeated until a predefined number of the accepted realizations converge.

Selecting the most accurate realization $m$, or a set of them, directly controls the quality of the generated 3D realizations, and the rate of convergence to a completely plausible model. One can use the 2D DIs to build the 3D realizations, but this increases the CPU time. Thus, as mentioned earlier, we propose an alternative process that uses a multiple-point histogram to construct a pattern distribution. Then, the distance $d$ of each model $m$ to the DI is quantified using the Jensen–Shannon (JS) divergence (Endres and Schindelin, 2003), which represents a method for measuring the similarity between two probability distributions. This data mining approach is the average of two kullback-leibler divergences,

$$d_{JS}(m) = \frac{1}{2} \int_{m}^{\infty} \log \left( \frac{D_{m}}{m} \right) dx + \frac{1}{2} \int_{0}^{m} \log \left( \frac{m}{D_{m}} \right) dx$$

(7)

and is always positive. Finally, a set of realizations that best maximizes the pattern distribution is selected; see Fig. 2. Altogether, by using the above described smart sampling, the number of samples that should be generated and, consequently, evaluated decreases dramatically. The algorithm is summarized in Algorithm 1.
3. Results and discussion

The algorithm proposed in this paper may be used when several 2D DIs and various types of experimental data are available. To better demonstrate the described method, however, we used a single 2D gray-scale DI and the measured permeability data. It should be noted that the initial gray-scale images were first segmented into a binary image for representing the pores and grains (Andrá et al., 2013) for the permeability calculations. In other words, the segmentation is performed on the generated stochastic models. We then used a simple thresholding scheme to discretize the images, which was used in the computations. It should be noted that other segmentation methods, such as the watershed and clustering-based methods, were also implemented (Karimpouli and Tahmasebi, 2016). In this study, however, the manual thresholding scheme was used as the segmentation method. For this aim, based on our knowledge of pore topology of shale samples, we defined a series of proper thresholds by which the gray scale images were converted into binary media. Due to using one single image as the input, the demarcated thresholds were utilized for the rest of modeling.

Before generating the stochastic pool of the 3D realizations, one must determine the optimal sample size, namely, the so-called representative element volume (REV). Various methodologies for similar shale samples are discussed elsewhere (Tahmasebi et al., 2016). For example, using a very small sample generally results in increased calculated permeability. On the other hand, by increasing the sample size the calculated permeability tends to converge to a stationary value. Thus, the selected size of the stochastic realization should not be too small to overestimate the permeability and, in a similar fashion, it should not be very large to increase the computational time. Furthermore, numerous lower- and higher-order tests, including variogram reproduction, marginal distribution and flow properties were carried out in our previous studies and they all produced good agreement with the experimental and original datasets (Tahmasebi et al., 2016).

Algorithm 1. Digital stochastic shale characterization.

1: Initialize $m_{\text{aux}} \leftarrow \text{CCSIM(DI)}$
2: Initialize $AUX_{\text{aux}} \leftarrow \text{movingAverage(m}_{\text{aux}})$
3: Initialize $SD_{\text{aux}} \leftarrow \text{mean(DI)}$
4: Initialize $\text{mutStep}$
5: $u \sim U(0, 1)$
6: for iteration $n = 1, 2, \ldots, \text{do}$
7: generate $m_{\text{aux}} \leftarrow f(q(m_{\text{aux}}) | SD, AUX, DI)$
8: $\rho(m_{\text{aux}}, m_{n-1}) = \min \left( \frac{1}{\text{exp}^{q^2(\mu_{m_{n-1}} - \mu_{m_{aux}})^2}}, C_2^2(f(m_{n-1}) - f(m_{aux})) \right)$
9: if $u < \rho$ then
10: $j = j + 1$
11: $m_{\text{aux}} \leftarrow m_{\text{aux}}$
12: if $(n \% \text{mutStep} == 0)$
13: $p(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(x-\mu)^2}{2\sigma^2} \right)$
14: end if
15: else
16: $m_{\text{aux}} \leftarrow m_{n-1}$
17: end if
18: Update the SD
19: Update the AUX
20: Calculate the JS divergence and update the DI as illustrated in Fig. 2
21: end for

*a, \mu, \sigma$ represent the mean, standard deviation and variance.

Fig. 2. — Selecting the most accurate pattern distribution, or a set of them, as the DI. In this case the most accurate model is the fourth from the left. However, a combination of the second and sixth models yields the most accurate pattern distribution. The objective is to identify those models that, if combined, offer a distribution that contains most of the pattern’s configurations.

3.1. Two-dimensional synthetic shale

The proposed algorithm begins by using the supplied DI, AUX, and SD. For the initial iterations, the marginal probability of the DI is

![Figure 3](image3.png) Fig. 3.—(a) Reference (true) generated using the CCSIM. (b) An initial stochastic realization. The size of the images in (a) and (b) are $150 \times 150$ pixels ($5 \times 5 \text{\mu m}^2$).
where \( p \) is the pressure of the fluid, \( u \) is its velocity, and \( \mu \) is the viscosity, taken to be 0.001 Pa·s. We assumed that the fluid is Newtonian and incompressible, and the flow is creeping under steady-state conditions. The downstream and upstream pressures were assumed to be 100 kPa and 130 kPa, respectively—equivalent to a pressure gradient of 3 kPa/\( \mu \)m. The permeability was computed using a finite-volume scheme based on the Avizo, which simplifies the computations of flow field. After the flow field was computed, Darcy’s law was used to compute the effective permeability.

The error of the computed permeability is compared with that of the previous realization. The current model is accepted if its error is less than the previous realization’s, and rejected otherwise. A weighting step, as indicated by Eq. (6), is added to the algorithm and an appropriate weight is assigned to each accepted realization. After a few iterations, the accepted realizations create a pool, the ensemble average of which reveals very useful information. A few members of the pool are shown in Fig. 4. Then, one uses such realizations and put the algorithm in a training workflow to produce other realizations with smaller error in the absolute permeabilities. To do so, the ensemble average map of the accepted realizations is used as the SD that conditions the subsequent realizations. The SD here represents the probability of occurrence of organic matter. Using the algorithm, we update the model pool, SD, and AUX data; see Fig. 5. There is no need to update the DI as we already have a comprehensive 2D image. This step is further discussed in the next example.

One may argue that the future realizations may mimic the most successful (i.e. matched) previous realization with a small perturbation, since the algorithm is conditioned to the SD. To avoid such a problem, a useful concept of the GA, the mutation step, is customized here. Thus, the weight of the accepted realization is changed after specific intervals. In fact, the SDs are generated using a combination of the weights and accepted realizations. For this reason, a Gaussian distribution is used to reassign the weights. Indeed, mutation breaks the previous probability in the SD and discovers more scenarios. In other words, as the SD leads the algorithm to converge on a shale-sample model, mutation provides an opportunity for exploration and testing more models in the search space.

The results of the proposed method are summarized in Fig. 6 in which a sequence of various stochastic models along with their corresponding SD and AUX data are depicted. As can be seen, the initial SD and, consequently, the AUX data represent a very smooth map caused by the high uncertainty in the initial models. As the algorithm proceeds, however, the SD and AUX take on more structure and become less smooth, and gradually converge to the model represented by the reference case in Fig. 3(a). It is worth mentioning that after a certain number of iterations, the generated stochastic models (the SD and AUX data) do not change significantly. As is shown in Fig. 6, in this example 200 realizations were generated, each of which took 2 CPU s. Therefore, the algorithm was stopped when the variance of a few consecutive models was close to zero.

3.2. Three-dimensional real shale-gas system

The second example represents a more complex case in which a 2D image is used to construct 3D realizations. A series of experiments were carried out on the sample. Nitrogen sorption (Clarkson et al., 2013; Hu et al., 2015) was measured on a powder size of 297–841 \( \mu \)m using a cleaned sample and under no confined stress. Then, the helium porosity test (Javadpour and Ettehadtavakkol, 2015) was implemented at 30 °C that yielded a porosity of 3.9 ± 0.3%. Finally, pulse decay permeability was performed at a confined stress of 13.8 MPa at the same temperature that yielded an absolute permeability of 400 nD (Bhandari et al., 2015). Thus, a 3D realization was generated using the CCSIM algorithm; see Fig. 7. Then, forward modeling in the form of flow calculations was carried out to estimate the absolute permeability. Next, the error, the difference between the computed permeability and the experimental value, was calculated. The generated realization is accepted if the result computed by Eq. (5) is less than 1 and rejected otherwise. Each new realization is accepted if its error (in the permeability) is less than that of the previously accepted realization. The accepted realizations are all placed into a model pool and used for various purposes. The database is then used to produce 3D SDs that are used to guide the algorithm and condition new realizations. As mentioned earlier, however, to differentiate the effect of the accepted realizations on the SD, a weight is assigned to each of them. The realizations with lower weights do not contribute much to the SD and, consequently, to the subsequent 3D models. Then, the SDs are generated using the accepted realizations and their relative weights. The number of the accepted realizations increases as
the algorithm proceeds, yielding more accurate conditioning data, i.e., the SD. Furthermore, the AUX data must be updated accordingly. Thus, the initial rough SD and AUX become more accurate as the algorithm converges and the models’ computed permeabilities draw closer to the experimental data.

Two methods are suggested in this paper for generating 3D realizations. One is using the initial 2D DIs, generating a 3D model, and proceeding to other steps of the workflow as before. The computation time of the method is very high, however, due to the direct 2D-to-3D reconstruction. The second approach is to generate a few acceptable 3D realizations, select a set of the most informative ones, and use them as the DI. Thus, 2D DIs are replaced with 3D ones, and the workflow becomes 3D-to-3D modeling, which decreases the computation time tremendously.

The use of iteration in this study is illustrated in Fig. 8. The SD possesses a very smooth structure at the beginning, indicating that...
organic matter is distributed widely. The absolute permeability changes from 1200 to 500 to 401 nD after 1, 147, and 285 iterations (realizations), respectively. Altogether, the CCSIM model, patchiness removal and histogram matching took 25 CPU s per realization. It should be noted that the permeability calculations for the 3D samples were performed on graphical processing units (GPU), which result in a very fast evaluation. In this example, the permeability computations took about 4 GPU min on a GeForce GTX 750Ti NVIDIA card. As the algorithm proceeds, the SD and, consequently, the AUX data become more accurate and less smooth. For example, the last 3D SD contains the most probable locations of the pore networks in the 3D sample. Note that the proposed method, regardless of the pore-size distribution of the original sample, can be used for any multi-scale structures that are very common in shale formations (Lemmens and Richards, 2013).

For the sake of comparison, pore-throat size of the original sample is compared with the produced realizations (Münch and Holzer, 2008; Sahimi, 2011). First, an equivalent binary image $B(p)$ is produced based on the generated realization, in which all the pores’ locations are identified, $P \subset B$. The grain location is denoted by $\mathcal{T}$, where $\mathcal{T}_{\cup P} = B$. The distance map $d(p) = \min |p' - p|$ is calculated, which represents the closest distance of the pore space to its grain boundaries. The map represents the possible locations of the pores where the centers of all spheres of radius of $r_s$ are located. The comparison is shown in Fig. 9. Although the realizations do not reproduce the exact pore-throat size distribution of the original sample, they represent a cloud of variation around the true distribution. This variation indicates the fact that several samples with different pore-throat size distributions can produce similar absolute permeabilities. A bimodal distribution of the pore-throat size is evident in Fig. 9, indicating a complex distribution of pore space in shale reservoirs (Shultz, 2015). Furthermore, the fraction of the organic matters and their associated porosity are also calculated for both the original and the realizations. The average of original materials in the original sample and 3D realizations were 19.50% and 21.27%, respectively. The porosity associated with each of the original sample and generated realizations was also computed to be 2.55% and 2.70%, accordingly. Furthermore, the TOC were calculated to be 55% and 56% in the original and the realizations, correspondingly, both indicating very good agreement with the data.

4. Conclusions

Producing equi-probable 3D shale realizations validated with measured permeability is of high importance. In our study, we introduced

---

Fig. 8.—Various 3D models and their corresponding SD. The SD represents the probability of occurrence of the organic matter in the model, and iteration 285 shows the locations where such features most probably exist. The size of the images is $600 \times 600 \times 600$ pixels ($9 \times 9 \times 9 \mu m^3$).

Fig. 9.—Comparison of pore-throat size distribution of the original sample (black) and those of several 3D realizations (gray).
a new iterative method that takes a few 2D images of a complex shale sample and reconstructs its 3D model stochastically. Such reconstruction, which uses physical rules derived from the given DIs, produces realizations that contain the pore networks in the actual samples. Due to its direct usage of the SEM images in the stochastic realizations, the reconstructed 3D models forecast more accurately the TOC of shale reservoirs than any of the previous methods.

As is well known, stochastic methods must be conditioned further to produce certain important properties. For example, experimental data for such properties as flow modeling and electrical conductivity can be integrated to reduce the uncertainty and produce more realistic realizations. Our approach for reconstructing 3D samples uses a high-order statistical approach represented by a cross-correlation function that efficiently integrates the experimental data. The method is based on the Metropolis-Hasting approach and the genetic algorithm, which are cast on an iterative learning concept in such a way that the previous iterations produce the next realizations.

Finally, the approach proposed in this paper can be used to better understand the complexity of shale reservoirs that produce gas. The method generates realization of shales that match the measured permeability without using full 3D imaging. Clearly, providing a number of such realizations improves our knowledge of the pore-network connectivity of shales. The method may also be used with various types of experimental data, such as the permeability, electrical conductivity. Clearly, more experimental data reduce the uncertainty in the realizations. The proposed method in this paper, regardless of the pore structures, can be used for any type of natural porous media with multi-scale structures.

Acknowledgements

Work at USC was supported by Research Partnership to Secure Energy for America (RPSEA) (53–4504-1115), and at the University of Texas at Austin by the NanoGeosciences Laboratory and Mudrock Systems Research Laboratory (MSRL) consortium. Dr. R. Loucks provided the SEM images. Publication was authorized by the Director, Bureau of Economic Geology. All data and digital content in this manuscript can be accessed by sending an e-mail to ptahmase@uwyo.edu.

References


